NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MAXIMUM LIKELIHOOD DECODING

AND BURST ERROR CORRECTION

by

Charles Alexander Davison

Thesis Advisor: John M. Geist

June 1972

Approved for public release; distribution unlimited.

MAXIMUM LIKELIHOOD DECODING AND BURST ERROR CORRECTION

Charles Alexander Davison

Naval Postgraduate School
Monterey, California

June 1972

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Maximum Likelihood Decoding and Burst Error Correction

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; June 1972

5. AUTHOR(S) (First name, middle initial, last name)

Charles Alexander Davison

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1972	46	9

8a. CONTRACT OR GRANT NO b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A brief discussion of basic encoding and decoding on noisy channels is presented to provide a background for the experimental portion of this research.

A partitioned 3 state Gilbert model is used to model a burst channel and a method of calculating error sequence probabilities using this model is introduced.

Error sequence probability calculations are made using a (7,3) maximal length code and a (15,7) BCH code.

Observations are made about the general type of decoding rule to use to give the lowest probability of decoding error on burst channels when using an interleaving technique.

I

DD FORM 1 NOV 68 1473 (PAGE 1)

/N 0101-807-6811

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

II

DD FORM 1 NOV 68 1473 (BACK)

S/N 0101-807-6821

Maximum Likelihood Decoding

and Burst Error Correction

by

Charles Alexander Davison
Lieutenant, United States Navy
B.S., Purdue University, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1972

Author: Charles A Davison

Approved by: ______________________ Thesis Advisor

______________________ Chairman, Department of Electrical Engineering

______________________ Academic Dean

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. THE COMMUNICATION SYSTEM

1. General

A general communication system (Fig. 1) contains an information source, an encoder, a channel, a decoder, and a destination. The information source selects a desired message from a set of possible messages. The source encoder, when one is used, compresses the data by removing inherent redundance in the source output so as to make each possible output equally likely. The channel encoder reintroduces redundancy into the data to improve the reliability of transmission over the channel.

The communication channel is the medium of conveyance of information from the source location to the destination location. The channel decoder uses the redundance introduced by the channel encoder to correct errors introduced during transmission.

If a discrete memoryless source whose output is equally probable binary digits is assumed, the source encoder and source decoder are no longer necessary and the resulting communication system is shown in Figure 2. In actual practice this is the most common arrangement for the communication system even if the information source does not produce equally probable binary digits.

If a communication channel is noisy, it is not possible, in general, to reconstruct with certainty at the channel decoder, the output of the information source. Shannon [1], however, did show that by proper encoding the probability of making a decoding error can be made arbitrarily small if the rate of data transmitted across the communication channel does not exceed a maximum value known as the channel capacity C.

The capacity of a channel, in general, is influenced by a number of factors. The number of channel inputs and outputs, and the set of all possible transition probabilities from the inputs to the outputs, all affect the channel capacity.

2. Noise

The effect of channel noise is to introduce the possibility that the output of the channel may differ from the input to the channel. The particular way in which the noise affects the channel's input is determined by the type of channel and the type of channel noise encountered. Memoryless channels, which are often used as theoretical models, assume that all digits transmitted over the channel are affected independently by channel noise. Unfortunately, this memoryless property is rarely found in real channels, an important exception being certain deep space channels.

Errors on most real channels tend to occur in groups or bursts. These real channels are thus channels with memory because the probability of the channel changing a transmitted digit is dependent on

whether the channel changed the previously transmitted digit. The calculation of the probability of a given error sequence occurring is thus the product of a series of conditional probabilities.

B. NOISY CHANNEL ENCODING AND DECODING

If noise were not present in the channel, no encoding of the source output would be needed to get the transmitted message to its destination. The presence of noise, however, requires sufficient redundancy in the encoded message so that the original message can be recovered at the decoder.

For binary encoding this required redundancy can be accomplished by using block codes and partitioning the input sequences into blocks of K bits. The encoder outputs blocks of a longer length (N bits) forming a (N,K) block code. The encoder thus maps the set of 2^K possible K bit sequences (messages) into a set of N bit sequences called code-words. In the channel, noise may be present and the input to the channel decoder (Y) may differ from the output of the channel encoder (X). The decoder performs the mapping of all possible received sequences back into the messages most likely to have been transmitted. Since the decoder must make a decision as to which message was transmitted for a given received sequence, there is a certain probability of making a decoding error. The probability of the decoder making an error is largely dependent on the mathematical properties of the type of code used, the type of decoding used, and the number and type of channel errors encountered.

II. CHANNEL MODELS

THE BINARY SYMMETRIC CHANNEL

The simplest model, and the one most commonly used to represent error sequences, is the binary symmetric channel. The BSC is shown in Figure 3. The transition probabilities are assumed to be constant and are not dependent on the previous uses of the channel. The maximum rate at which information can be reliably transmitted across the channel is called the channels capacity C. Shannon [1] and others have shown that the capacity of the binary symmetric channel $C = 1-H(p)$, where the p is the crossover or error probability of the BSC as shown in Figure 3. H(p) is called the entropy function and is defined as $H(p) = -p \log_2 p-(1-p) \log_2 (1-p)$.

In information theory, the binary symmetric channel is the most often used model of a communication channel. This idealized model has been shown to accurately represent some deep space communication links but it is a poor model for most real communication channels encountered. Errors on real channels caused by lightning interference from another transmitter, fading propagation paths, and many other natural and man-made phenomena tend to occur in groups or bursts.

Bursty channels are called channels with memory because the probability of making an error on a particular digit of an information sequence is greatly increased if an error is made on the preceding digit. The memory

characteristic of most real channels is the reason the BSC fails as an accurate model.

B. THE GILBERT MODEL

1. Introduction and Description

A simple model for a channel with memory was proposed by E. N. Gilbert [5] . The pictorial representation of the Gilbert model is shown in Figure 4. This model of a burst-noise binary channel uses a Markov chain with two states called G and B. In state G no transmission errors are made and in state B the probability of making an error is h. The parameters h, P, and p are assumed constant. The probability of making an error on any digit of a binary sequence is dependent on the state of the channel when that digit is transmitted. The fraction of time spent in the bursty state B, is $p/(P+p)$ and the fraction of time spent in the good state is $P/(P+p)$.

2. The 3 State Partitioned Gilbert Model

a. Definition

In the two state Gilbert Model when in the bursty state B an error may or may not occur. If the bursty state B is partitioned into an error free state B_0, and an error state B_1, the resulting channel model is as shown pictorially in Figure 6. The model now consists of three states: G, B_0, and B_1, and errors occur when and only when the channel is in state B_1.

With this partitioned model, the probability of a digit of a transmitted binary sequence being in error is dependent on the state of the channel and the transition probability from that state to the B_1 state. The state of the channel is determined by the previously transmitted digit. If the previous digit was in error, the channel is in state B_1 and the probability of a digit error $P_n(E) = h(1-P)$. If the previous digit was not received in error, the channel could be in either state B or state G and $P_n(E) = P_n(B_0/\text{NOT } B_1)\, h\,(1-P) + P_n(G/\text{NOT } B_1)\, P$ WHERE $P_n(B_0)$ is the probability of the channel being in state B_0 and $P_n(G)$ is the probability of the channel being in state G. In a similar manner it is possible to calculate the probability of occurrence of a complete binary error sequence, using this model. The probability of a digit error is the calculation of the probability of the state B_1. To calculate the probability of an error sequence, the a priori state probabilities must be calculated for the given model parameters P, p, and h.

b. Calculation of a priori State Probabilities

Since the probability of occupancy of a state at any digit is only determined by the previous digit and the transition probabilities, the state probabilities at the (k+1) ST digit can be expressed by the following equations:

$$\pi_G(k+1) = (1-p)\,\pi_G(k) + P\,(\pi_{B_0}(k) + \pi_{B_1}(k))$$

$$\pi_{B_0}(k+1) = p(1-h)\,\pi_G(k) + (1-P)(1-h)\,(\pi_{B_0}(k) + \pi_{B_1}(k))$$

$$\pi_{B_1}(k+1) = hp\,\pi_G(k) + (1-P)\,h\,(\pi_{B_0}(k) + \pi_{B_1}(k))$$

As $k \to \infty$, the probabilities $\pi G(k)$, $\pi B_0(k)$, and $\pi B_1(k)$ approach equilibrium values πG, πB_1, and πB_0. Thus, the equations reduce to:

$$\pi_G = (1-p)\,\pi_G + p\,(\pi_{B_0} + \pi_{B_1})$$

$$\pi_{B_0} = p(1-h)\,\pi_G + (1-p)(1-h)(\pi_{B_0} + \pi_{B_1})$$

$$\pi_{B_1} = hp\,\pi_G + (1-p)\,h\,(\pi_{B_0} + \pi_{B_1})$$

Since $(\pi_{B_0} + \pi_{B_1}) = (1 - \pi_G)$ it's substitution yields

$$\pi_G = \frac{p}{p+p} \qquad \pi_{B_0} = \frac{p(1-h)}{p+p} \qquad \pi_{B_1} = \frac{hp}{p+p}$$

An alternate method of solution is to observe in the original Gilbert model that the fraction of time spent in state G is P/(P+p) and the fraction of time spent in the burst state B is p/(P+p). The fraction of time spent in state B_1 is the burst state error probability h times the fraction of time in the burst state and the time spent in state B_0 is thus (1-h) p/(P+p).

c. Calculation of Error Sequence Probability

Given any binary sequence, the probability that that sequence is the error sequence of a Gilbert channel can be calculated as follows:

(1) Initialize by choosing values $\pi G(0)$, $\pi B_0(0)$, $\pi B_1(0)$. (If the initial state of the channel is unknown or not specified, a sensible choice is to initialize to the state equilibrium distributions: $\pi_{\bullet}(0) = \pi_G$, etc.)

(2) If the k th digit is in error, assign

$$\pi_{B_0}(k) = \pi_G(k) = 0.0$$

$$\pi_{B_1}(k) = h \cdot \pi_G(k-1) + (1-p)h\left(\pi_{B_0}(k-1) + \pi_{B_1}(k-1)\right)$$

Assign the probability of the sequence after the k th digit PSEQ(k) = $\pi B_1(k)$

(3) If the k th digit is not in error, assign $\pi B_1(k) = 0$

$$\pi_{B_0}(k) = p(1-h)\pi_G(k-1) + (1-p)(1-h)\left(\pi_{B_0}(k-1) + \pi_{B_1}(k-1)\right)$$

$$\pi_G(k) = (1-p)\pi_G(k-1) + p\left(\pi_{B_0}(k-1) + \pi_{B_1}(k-1)\right)$$

Assign the sequence probability after the k th digit PSEQ(k) = $\pi B_0(k) + \pi G(k)$

d. Modeling of Real Channels

In recent years many channel models have been proposed to characterize the performance of real communications channels. Gilbert [5] originally proposed the simple two state model for a channel with memory and had limited success in choosing the parameters of his model to produce statistics similar to given finite length error sequences. Using this model it is impossible to reconstruct the sequence of states from a given error sequence because of the many possible sequences of states that produce the same given error sequence.

Fritchman [7] extended the model of Gilbert by studying the general case of finite state models with k error free states and N-K error states. Many more complex models have been developed in attempts to accurately represent the performance of real channels. The comparison of the accuracy of a developed model to a given real channel is usually done by performing a statistical analysis on a finite data

sequence from the channel and comparing the results to the statistics of data produced using the constructed models. Increasing the model's complexity increases the number of possible ways the model can generate a particular sequence and thus reduces the chance of obtaining accurate statistical data about the model.

III. CODING ON A NOISY CHANNEL

A. BLOCK CODES

1. General

Block codes are usually specified as (N,K) codes, where N is the number of codeword digits (block length) and K is the number of information digits in a codeword. The rate, R, of the code is the ratio of the number of information digits in a codeword to the total number of digits in the codeword (R= K/N). The hamming distance between two codewords is the number of positions in which the digits of the two codewords differ. The hamming weight of a codeword is its number of non-zero components. The distance between two binary codewords is the hamming weight of their difference. The distance between a transmitted codeword $\underline{x}_m$ and the received codeword $\underline{y}$ denoted $d(\underline{x}_m, \underline{y})$, is the number of transmission errors occurring in the channel.

2. Error Correction Bounds for (N,K) Block Codes

a. Random Error Correction

Let d_{min} denote the minimum distance of a (N,K) block code (the least hamming distance between codewords). At least two codewords differ in only d_{min} of their N positions. It has been shown by Peterson [2] and others that these block codes with minimum distance d_{min} can in general detect $d_{min} - 1$ errors or correct $(d_{min}-1)/2$ errors. It is also possible to decode in such a way as to

simultaneously correct t or fewer errors and detect m or fewer errors if and only if $d_{min} > 2t + m$. If a code is used for error correction only and has a minimum distance of d_{min}, $d_{min} > 2t$, the code is capable of correcting t or fewer random errors.

Two codes can have the same miminum distance and one of the codes have much better error correcting ability because of its capability to correct more error patterns of greater weight than that guaranteed by its minimum distance. Minimum distance alone is therefore not a complete measure of the goodness of a code.

Gilbert [8] proved that for any $N > 0$ and $d > 0$ such that $\frac{d}{N} \leq \frac{1}{2}$, there exists a code of length N and minimum distance $d_{min} \geq d$ with a rate $R \geq 1-H(d/N)$ (where $H(\ )$ is the binary entropy function). This bound, known as the Gilbert Bound, is often used as a measure of goodness for a code. Since $t = \frac{d-1}{2} < \frac{d}{2}$ errors can be corrected by a code with a minimum distance d, the Gilbert bound may be expressed as

$$H\left(\frac{2t}{N}\right) \geq 1-R$$

b. Burst Error Correction

An error sequence of length N is said to contain a burst of length t if all non-zero digits are confined to a span of t consecutive positions. Since a burst of length t is also one of the random error patterns of weight t, it is clear that a code capable of correcting any pattern of t or fewer errors is also capable of correcting all bursts of

length t or less. Gallager [3] has shown that a code of length N and rate R can correct all bursts of length t or less only if

$$\frac{t}{N} \leq \frac{1}{2}(1-R)$$

This relation is known as the Gallager bound and it can be shown that as the block length N approaches infinity, codes exist which meet the Gallager bound.

c. Comparison of Burst Correction and Random Error Correction

A comparison of burst and random error correcting capabilities of codes can be obtained by comparing the Gilbert and Gallager bounds as shown in Figure 6. Also sketched is the asymptotic form of an upper bound on random error correcting capability due to Plotkin [9]. The bounds show that as N approached infinity, the length of correctible bursts is twice the weight of correctible random error patterns.

B. LINEAR CODES

1. General

The alphabet of two symbols, 0 and 1, under modulo-2 addition and multiplication is called the Galois field of two elements (or binary field) and is usually denoted GF(2). It can be shown that for any integer $q = p^n$, where p is prime and $n \geq 1$, a Galois field of q elements exists. This field is usually denoted GF(q). The set of all binary N-tuples is a vector space over GF(2) of dimension N under the operation of modulo-2 addition. A binary code is called linear if and only if it is a subspace

of the space of all N-tuples. Any linear combination of codewords of a linear code is thus also a codeword of the linear code. Since any codeword added to itself is a codeword, the N dimensional null vector is always a codeword of any linear code.

2. Generator Matrix G and Parity Check Matrix H

Any set of basis vectors for a linear codeword set V can be considered as rows of a matrix G called the generator matrix. All codewords are linear combinations of the rows of G. If the dimension of V is K, the number of rows of G is K and G if a (KxN) matrix. Every codeword $\underline{x}$ in the codeword set V can be generated by multiplying the matrix G by the vector $\underline{u}$ where $\underline{u}$ is one of the set of 2^K K-tuples, called messages ($\underline{x} = \underline{u}G$).

The parity check matrix H for a linear code is a matrix such that for any $\underline{x}$, $\underline{x}H^T = 0$ if and only if $\underline{x}$ is in V. H is thus a ((N-K) x N) matrix of rank N-K.

A codeword set V is in canonic systematic form when the first K digits of a codeword $\underline{x}$ is the information vector $\underline{u}$ used to generate $\underline{x}$. The codeword $\underline{x}$ may be expressed by $\underline{x} = (a_1, a_2, \ldots . a_K, c_1, c_2, \ldots . c_{N-K})$. The G and H matrices can now be expressed by

$$G = [I_K \mid P] \qquad H = [-P^T \mid I_{N-K}]$$

(I_K denotes a identity matrix of order K). It can be shown that any linear code can be put in canonic systematic form after a proper permutation of its codeword positions.

3. Syndrome Decoding

The syndrome S of a linear code may be defined by $\underline{S} = \underline{y} H^T$ where $\underline{y}$ is the received sequence at the channel decoder. The received sequence $\underline{y}$ may be expressed as $\underline{y} = \underline{x} + \underline{e}$ where $\underline{x}$ is the codeword transmitted and $\underline{e}$ is the error sequence generated by additive noise in the channel. $\underline{S} = \underline{y} H^T = \underline{x} H^T + \underline{e} H^T = \underline{e} H^T$.

(since $\underline{x} H^T = 0$ for any codeword $\underline{x}$.)

Since $\underline{S} = \underline{e} H^T$ and $\underline{e}$ is a N component vector and H^T is a N x (N-K) matrix $\underline{S}$ is a vector with N-K components.

In general for any binary (N-K) linear code there are $2^{(N-K)}$ syndromes and each of these syndromes has 2^K possible error sequences for which the equation $\underline{S} = \underline{e} H^T$ is satisfied. If the decoder is constructed so that upon receiving an input $\underline{y}$, it calculates $\underline{S}$, then chooses the $\underline{e}^*$ which is the most likely of the 2^K possible error sequences of $\underline{S}$, maximum likelihood decoding can be implemented by adding the $\underline{e}^*$ to $\underline{y}$ to yield $\underline{x}^*$, the codeword most likely to have been transmitted.

If $\underline{S} = 0$, then the received sequence $\underline{y}$ is a codeword and if $\underline{S} \neq 0$, the received sequence is not a codeword. $\underline{S} = 0$ does not guarantee that no errors were made in transmission since $\underline{x} + \underline{e}$ could sum to a codeword but $\underline{S} \neq 0$ does guarantee that some errors did occur.

One alternative decoding method to maximum likelihood decoding is to calculate the syndrome S and if S = 0, accept the codeword as received and if $S \neq 0$, request a retransmission of the codeword. The

main disadvantages to this system are the additional reliable communication systems needed from the user back to the source and if a large number of errors occur, the data rate is greatly reduced and a large buffer may be needed to maintain symbol synchronization.

C. CYCLIC CODES

1. Introduction

a. Definition

As defined by Gallager [3], a cyclic code over GF(q) is a linear code with the special property that any cyclic shift of a codeword is another codeword. That is, if $(a_1, a_2, a_3, \ldots\ldots a_N)$ is a codeword, then $(a_N, a_1, a_2, \ldots\ldots a_{N-1})$ is also a codeword.

b. Generation of a Cyclic Linear Code

If a codeword $\underline{x} = (x_{N-1}, x_{N-2}, \ldots\ldots x_1, x_0)$ it may be represented by a polynomial over GF(q) (a Galois field of q elements).

$$x(D) = x_{N-1} D^{N-1} + x_{N-2} D^{N-2} + \cdots x_1 D + x_0 .$$

If x(D) is a codeword in a cyclic code (the coefficients form the letters of a codeword) then the remainder of D x(D) modulo $D^N - 1$ is also a codeword. Let g(D) be the lowest degree monic polynomial of degree m(m = N-K), which is a codeword. It has been shown that for any polynomial a(D) in GF(q) with degree at most K-1, a(D) g(D) is a codeword. The polynomial g(D) is called the generator polynomial of the cyclic code and all codewords contain g(D) as a factor. The set of codewords is the set of linear combinations of g(D) and its first K-1 cyclic shifts. Any (N-K) degree

monic polynomial over GF(q) that divides $D^N - 1$ can generate a cyclic code with K information digits and block length N.

The check polynomial h(D) for a (N,K) linear cyclic code is defined so that $g(D)\,h(D) = D^N - 1$ and h(D) is of degree K. With the check polynomial h(D) so defined, it may be shown that, as for the parity check matrix for any linear code, $\underline{x}H^T = \underline{0}$ if and only if $\underline{x}$ is a codeword.

2. Maximal Length Codes

a. Definition

A linear maximal sequence is a binary sequence generated by a linear shift-register generator which has the longest possible period for this generation method. The longest period, $L = 2^K - 1$, where K is the number of stages in the shift-register generator. A linear code whose codewords are maximal length binary sequences is called a maximal length code.

b. Generation

A linear shift-register generator consists of a basic shift register and modulo-two adders. The generator outputs a binary sequence that is based on its initial input and the feedback connections to the modulo-two adders. The binary sequence output of the register is of maximal length when the feedback connections are made in accordance with a primitive polynomial as is defined by Peterson [2]. The connections also correspond to the parity check polynomial h(D) described in the previous section.

As described by Gallager [3], given h(D), a minimum polynomial of degree m of a primitive element in any representation of $GF(p^m)$, a maximal length code of block length $N = p^m - 1$ can be generated by an m-stage shift-register encoder circuit as shown in Figure 6. One codeword of the code corresponds to the generator polynomial g(D) and the remaining codewords are generated by N-1 cyclic shifts of g(D).

A (7,3) maximal length code may be generated using a feedback shift-register whose connections correspond to a primitive polynomial of degree 3. The third degree primitive polynomial listed in Peterson [2], is 1 3 (octal representation) or 001011 (binary representation). This corresponds to $h(D) = 1 + D + D^3$ and the connections to a feedback shift-register to generate the (7,3) maximal length code are shown in Figure 7.

Maximal length codes are useful because they are easy to generate and have a large minimum distance for their block length. The (7,3) maximal length code has a block length of 7, a rate of 3/7 and a minimum distance of 4. This code is thus able to correct all single errors and many double error patterns.

3. BCH Codes

a. General

The Bose, Chandhari, and Hocquenghen (BCH) codes were first discovered in 1959. These codes are cyclic codes which have powerful error-correcting properties and for which relatively simple decoding algorithms exist. The BCH codes have become the most important and

widely used linear cyclic codes. Most examples of BCH codes are binary, but the alphabet can be elements from any arbitrary Galois field GF(q).

For these BCH codes it is possible to specify a block length N (usually $N = 2^m - 1$) and a minimum distance d($d < N$) and choose a generator matrix to produce a code with the specified length and distance. For lengths up to 1023, the BCH codes have rates which meet or exceed the Gilbert bound, although as N approaches infinity they fail to do so.

b. Generation

Suppose a block length of 15 (m = 4) and a minimum distance of 5 was desired (ability to correct two errors). For α a primitive element of $GF(2^4)$, the generator polynomial for this desired code can be calculated by taking the product of the minimum polynomials for d-1 consecutive powers of α. (Refer to Gallager [3] page 233 for a brief list of minimal polynomials.) Calculation of G(D) by this method yields

$$G(D) = (D^4 + D + 1)(D^4 + D^3 + D^2 + D + 1) = D^8 + D^7 + D^6 + D^4 + 1 .$$

Since the generator polynomial is of degree N-K, N-K=8, K=7, and the code is a (15,7) BCH code. A possible generator matrix for this code is a matrix whose first row is the code vector corresponding to the generator polynomial G(D). Since $G(D) = D^8 + D^7 + D^6 + D^4 + 1$, the first row of the generator matrix could be 000 000 111 010 001. The remaining K-1 rows of the generator matrix could be the K-1 or 7 cyclic shifts of the first row.

4. Interleaving

a. General

A simple and often used technique to combat burst errors on a channel is the use of an interleaver. The principle is to separate successive digits within a codeword by a certain time interval so that burst errors on the channel will not appear successively in the codeword. If the interleaving achieved a separation of B bits, a burst of B errors would cause one error to appear in each codeword. This technique distributes the channel burst errors in a pseudo-random manner and gives the decoder an opportunity to correct an otherwise uncorrectable burst error pattern, at the possible expense of making more decoding errors. The two most common interleavers are the block interleaver and the periodic (or convolutional) interleaver.

b. Block Interleavers

Block interleavers are the most common type of interleavers and the interleaving is usually accomplished by storing encoded codewords bits in the rows of a BxN matrix and then reading out these bits by columns prior to their transmission across the channel. This produces a separation of B bits between adjacent bits of the codeword when it transits the channel. The longer the degree of interleaving, the more storage required and the longer time delay from the encoding of a word until it is actually transmitted across the channel. The received bits are deinterleaved prior to their decoding.

c. Periodic Interleaver

A periodic (or convolutional) BXN interleaver achieves interleaving by arranging the codeword symbols in blocks of N and delaying the i th symbol in each block by (i-1) B time units. The delay is accomplished using a (i-1) B^1 stage shift-register clocked once every N symbol times, where $B^1 = B/N$.

At the receiver, symbols are reblocked in groups of N by the deinterleaver and the i th symbol in each block is now delayed by (N-i) B times units using a (N-1) B^1 state shift-register.

The result of this interleaving and deinterleaving is to delay all symbols by (N-1) B^1 time units and separate adjacent codeword symbols by B time units. A single channel burst of B or fewer time units will affect only one of the N deinterleaver output streams at a time.

IV. DECODING

A. GENERAL

1. The Decoding Problem

The basic problem of the block code decoder is to chose the correct codeword transmitted from the set of 2^K possible codeword N-tuples which could have been transmitted given that a certain N-tuple (y) was received. There are a number of possible ways the decoder could make the required choice. Two decoding methods are maximum likelihood decoding and minimum distance decoding.

2. Maximum Likelihood Decoding

Let

$$\underline{x} = [x_1, x_2, \cdots x_N]$$

denote a transmitted codeword and

$$\underline{y} = [y_1, y_2, \cdots y_N]$$

the N-tuple received by the decoder. Given a certain $\underline{y}_m$ has been received the maximum likelihood decoder chooses a $\underline{x}_{\hat{m}}$ one of the set of 2^K possible codewords such that the probability $P_n(\underline{y}_m / \underline{x}_{\hat{m}})$ is maximized. To accomplish the proper choice of $\underline{x}_{\hat{m}}$ the decoder must calculate the probability of $\underline{y}_m$ for each of the 2^K possible codewords which could have been transmitted. Since each of these 2^K probability calculations takes time, maximum likelihood decoding is not really practical for long codes.

The minimum-error probability decoding rule is the decoding rule which minimizes the probability of decoding error for a given message ensemble of codewords. For discrete memoryless channels the decoder minimizes the probability of error by choosing a $\underline{x}_m$ so that the probability of that $\underline{x}_m$ conditioned on the received sequence ($\underline{y}$) is largest. If all of the 2^K codewords are equally likely (which is usually assumed), it can be shown that maximum likelihood decoding is equivalent to minimum-error probability decoding.

3. Minimum Distance Decoding

Given any two binary N-tuples the distance between them is defined as the number of positions in which the two sequences differ. The distance between a transmitted codeword $\underline{x}$ and a received sequence $\underline{y}$ is therefore

$$d(\underline{x}, \underline{y}) = \sum_{i=1}^{N} \delta\ (x_i, y_i)$$

where

$$\delta = 1 \text{ if } x \neq y \text{ otherwise } \delta = 0.$$

As previously stated, the minimum distance is a rough measure of a code's error correcting and detecting ability. A minimum distance of d_1 guarantees the ability to correct at least $\frac{d_1 - 1}{2}$ errors. $(\underline{x}, \underline{y})$ is the number of errors that have occurred in the channel. For memoryless channels

$$\Pr\ (\underline{y}/\underline{x}) = \beta^{d(\underline{x}, \underline{y})}\ (1-\beta)^{N - d(\underline{x}, \underline{y})}$$

where Pr denotes the probability and β is the probability of a digit being in error. Since β is always assumed less then 1/2 the $\Pr(\underline{y}/\underline{x})$

increases as $d(\underline{y},\underline{x})$ decreases. The decoder which minimizes $d(\underline{y},\underline{x})$ always maximizes $P(\underline{y}/\underline{x})$. On the binary symmetric channel with equally likely codewords, the "minimum distance" decoder is equivalent to a maximum likelihood decoder.

B. DECODING ON CHANNELS WITH MEMORY

1. General

The concept of maximum likelihood decoding applies to channels with memory, as well as to memoryless channels. Maximum likelihood decoding, however, cannot be implemented on a channel with memory by using a minimum distance decoding rule. The usual decoding strategy for channels with memory is to map each syndrome into the error pattern which is the shortest burst that could cause that syndrome. Decoders which are optimum in this sense are known [3], but these decoders are optimum in the sense that they have the lowest probability of error only if a short burst is always more likely than a longer burst.

In order to obtain some quantitative evaluation of maximum likelihood burst decoding a simulation of two codes using a Gilbert channel model was performed as is described in the following sections.

2. (7,3 Maximal Length Code

A (7,3) maximal length code was constructed (as described in Chapter III) and a channel was modeled using the partitioned 3 state Gilbert model.

A (7,3) maximal length code was chosen because it has a short block length, relatively large minimum distance (4), and is easily constructed. The code has 16 syndromes and 8 error sequences are solutions to $\underline{S} = \underline{e}H^T$ for each syndrome. Since the minimum distance of 4 gave a capability to correct all single errors, all single error sequences and the zero error sequence were assumed to be the most likely error sequence for their corresponding syndrome, since no two of these correctable error sequences were in the same syndrome. Seven of the remaining eight syndromes contained 3 weight two error sequences. The remaining syndrome contained 7 weight three error sequences and the burst of length seven.

The parameters P, p, and h of the partitioned 3 state Gilbert model were varied and error sequence probabilities were calculated for the zero error sequence, the seven single error sequences, and the 64 error sequences of the remaining 8 syndromes, using the method described in Chapter II.

A maximum likelihood decoding error probability, P(E), was calculated by summing the probabilities of the most likely error sequence in each of the 16 syndromes and subtracting this cumulative sum from one. The results of these error sequence probability and maximum likelihood decoding calculations are as follows:

(a) The most likely error sequence for each syndrome was the burst pattern of minimum length. Error sequences of the same weight were not in general equally likely using the Gilbert model but were dependent on the parameters P, p, and h.

(b) The probability of decoding error, P(E), increased as the channel model was made more noisy by increasing p or h or by decreasing P (see Fig. 9, 10, and 11).

(c) The probability of error is influenced more by P, the probability of transition from the burst state, than by the parameters p or h of the Gilbert model (see Fig. 9, 10, and 11).

(d) Finally, a binary symmetric channel was modeled by letting P = 1-p and h = 1. With this model, all error sequences of equal weight were equally likely, and the most likely sequences were the ones of least weight. The probability of decoding error, P(E), increased as the model was made more noisy.

Since this code has a very limited number of syndromes, its ability to correct long bursts was limited. This suggested a code of longer length with greater burst correction capability should be investigated.

3. (15,7) BCH Code

A (15,7) BCH code was constructed as described in Chapter III. This code has 2^8 syndromes and 2^7 error sequences are solutions to $\underline{S} = \underline{e}H^T$ for each syndrome. Since it was impractical to calculate all possible error sequence probabilities as was done for the (7,3) maximal length code, another method had to be used.

An error sequence of interest was chosen. All error sequences which have the same syndrome as the chosen error sequence, were

generated by the addition (modulo 2) of this error sequence to every code-word of this (15,7) BCH code. The probability of each of these sequences was then calculated (as described in Chapter III) using the partitioned three state Gilbert model. The results of these calculations were as follows:

(a) Error sequences of equal weight are not in general equally likely and the selection of the Gilbert model parameters determines which one of the equal weight patterns is the most likely.

(b) The most likely error sequence for a particular syndrome is not always the error sequence of least weight or the burst error sequence of shortest length. Figure 12 shows the probability of three different error sequences as h, the burst state error probability, is varied. For values of h between 1.0 and .75, a solid burst of length 6 is the most likely sequence. For values of h between .75 and .39, a burst of length 5 is the most likely sequence. When h is less than .39, the minimum weight sequence of weight three is the most likely.

V. CONCLUSIONS

Although many better and much more complicated channel models exist, a bursty channel can be modeled using the simple partitioned 3 state Gilbert model. Error sequence probabilities can be easily calculated using this model. These calculations show for some codes error sequences of the same weight are not equally likely and burst error sequences of smaller length may not be as likely as longer burst error sequences.

The optimum burst decoder, as proposed by Gallager [3], which always chooses the burst error of smallest length as the most likely error sequence, is not optimum in the sense of having an error probability as low as a maximum likelihood decoder. A minimum weight decoder likewise is also not a maximum likelihood decoder for this model. This suggests that a decoder having the minimum probability of error for the bursty channel cannot be easily constructed.

If it is possible to model a real channel using a finite number of states involving a Markov chain, it is then possible to calculate error sequence probabilities and choose the type of decoder required to give the lowest probability of decoding error.

A common technique for combatting burst errors has been to use an interleaver to scatter burst channel errors in a pseudo-random manner. The rationale behind this technique is if the degree of interleaving is large enough, the burst errors will be sufficiently scattered so that the channel

can be treated as memoryless. Since the errors now seem to occur in a random manner, a code with good random correcting ability is sometimes used in conjunction with a minimum distance decoder. The channel burst errors are not purely random but are distributed systematically in accordance with the interleaver used. Purely random error correction does not use information contained between interleaved codewords about how errors are distributed. Interleaving does enable a code to correct otherwise uncorrectable long burst errors. This suggests that a better burst error correction technique would be to use an interleaver but also use a decoding rule which would use the information contained between interleaved codewords to aid in burst error correction.

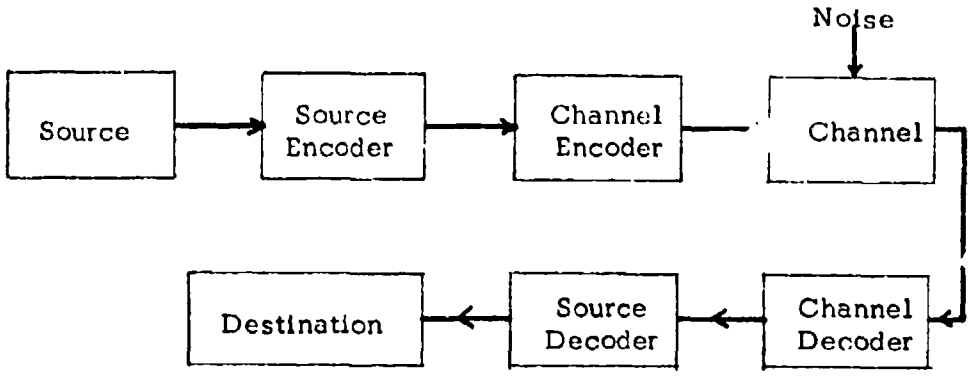

Figure 1. Block Diagram of a General Communication System.

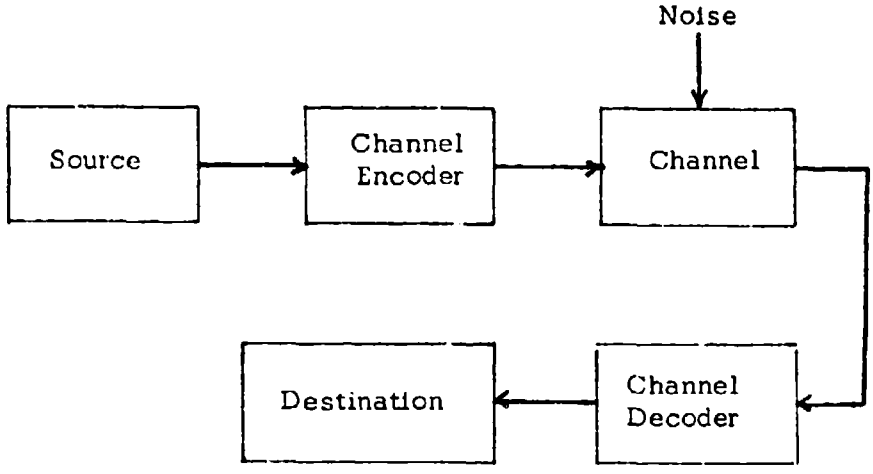

Figure 2. Block Diagram of a General Communication System with a Discrete Memoryless Source Assumed.

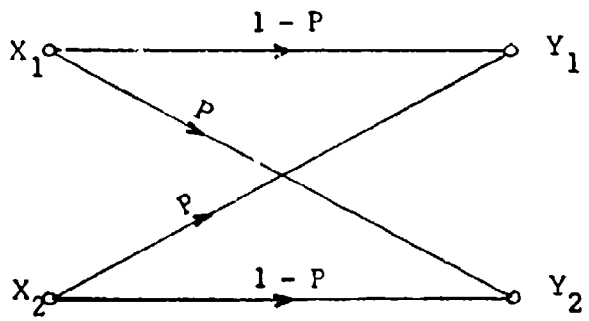

Figure 3. The Binary Symmetric Channel

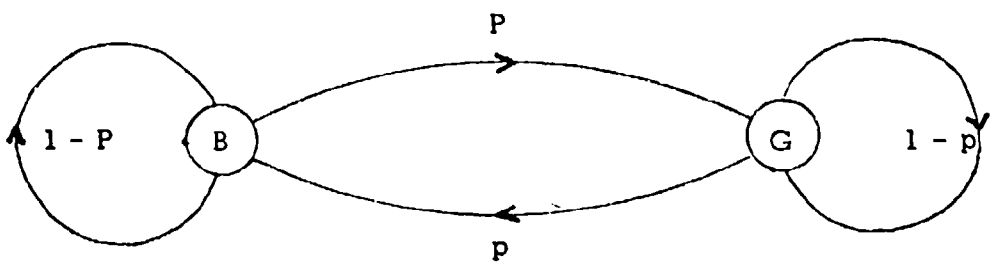

Figure 4. The Gilbert Model.

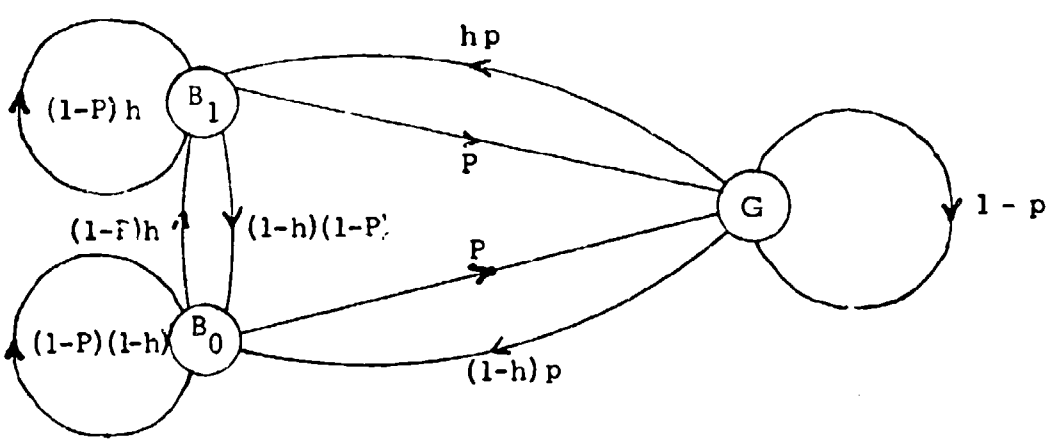

Figure 5. The Partitioned Three State Gilbert Model.

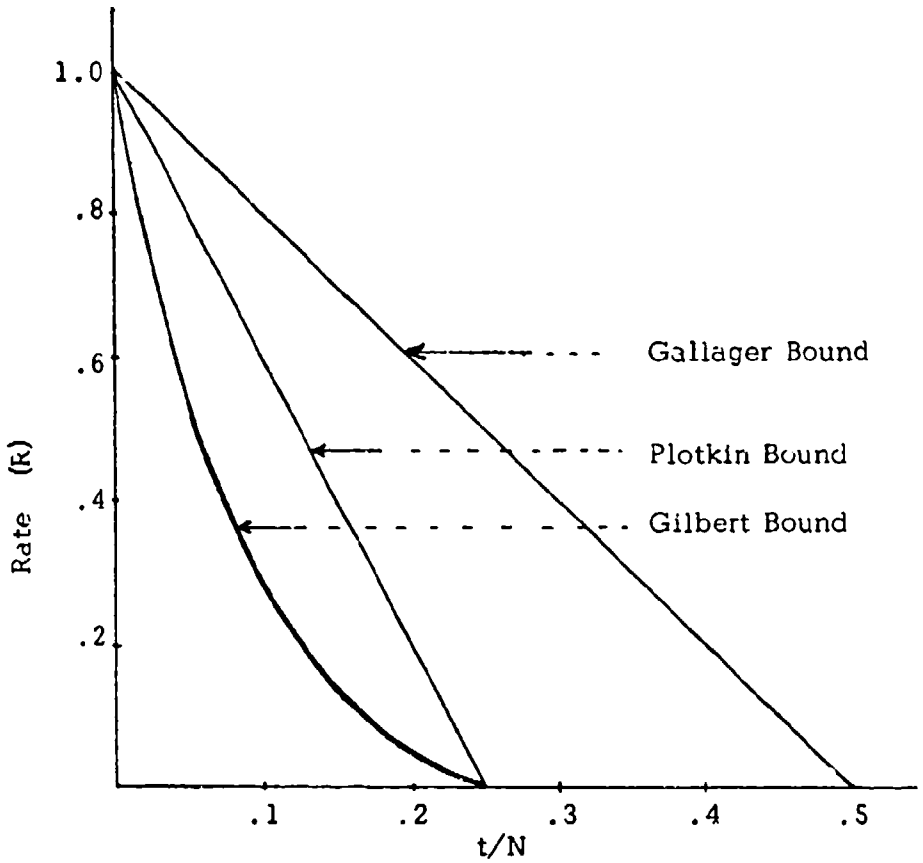

Figure 6. Asymptotic Error Correction Bounds.

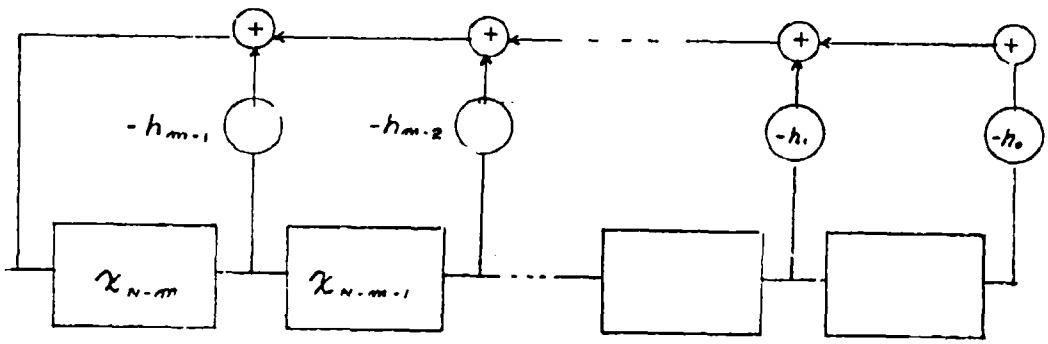

Figure 7. Encoder for a Maximal-Length Code.

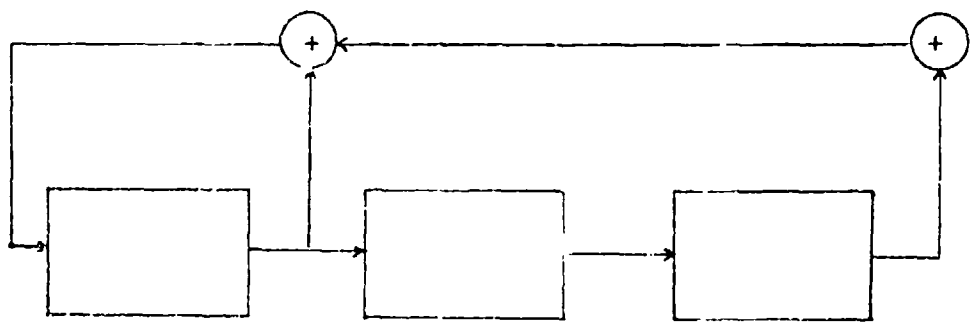

Figure 8. Feedback Shit Register Connection to Generate (7,3) Maximal Length Code.

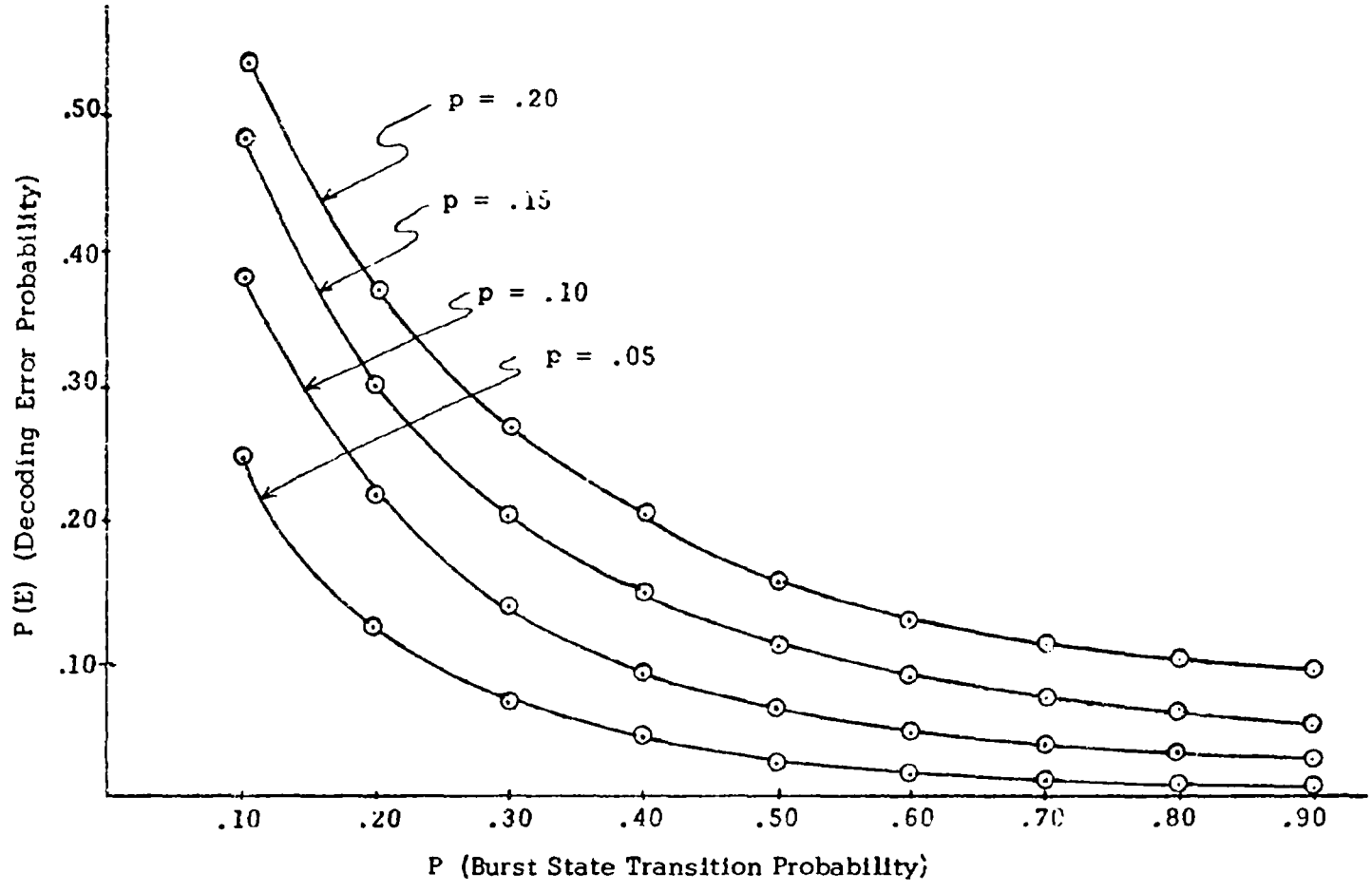

Figure 9. P(E) vs P (h = 0.5)

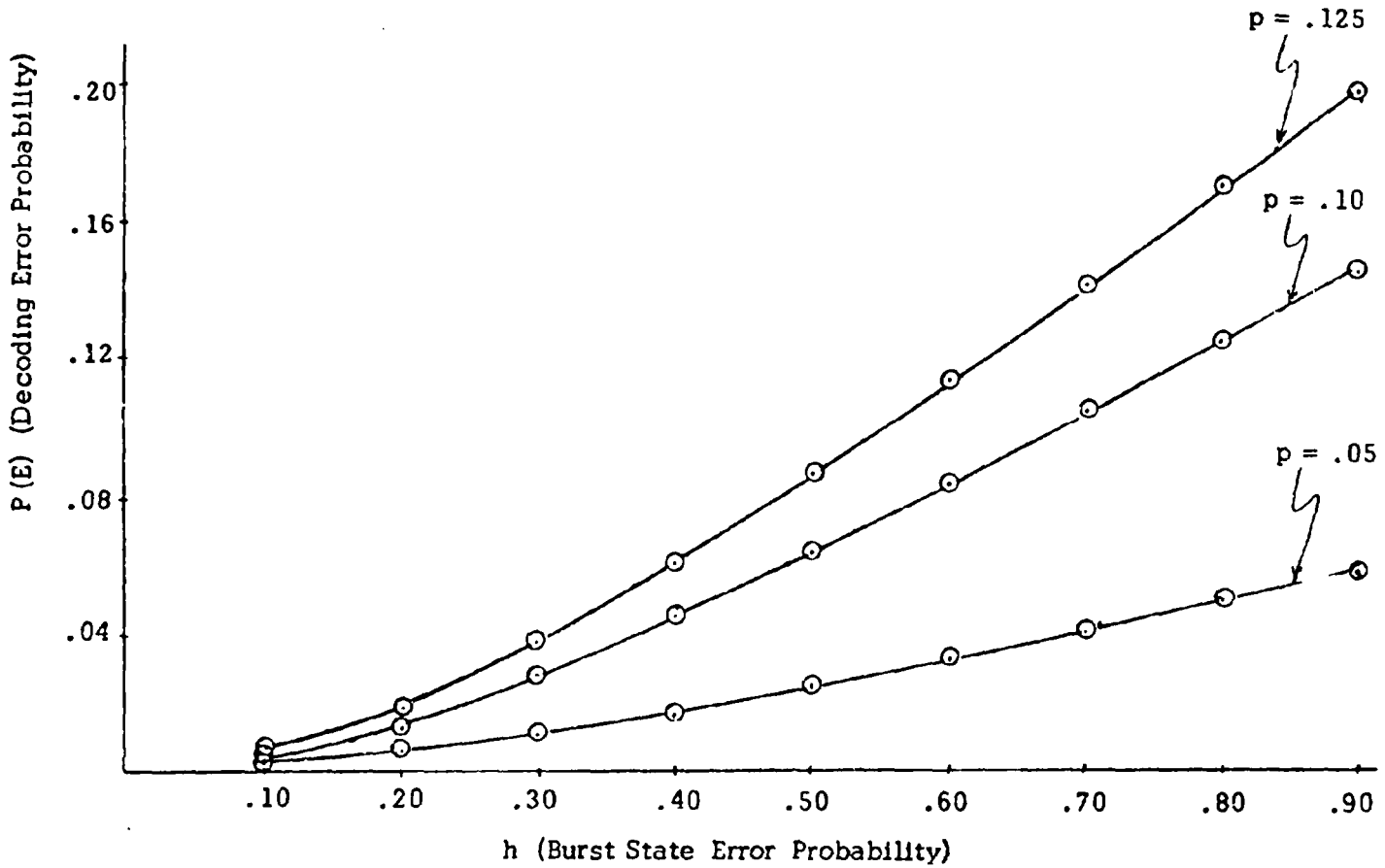

Figure 10. P(E) vs h (P = 0.5)

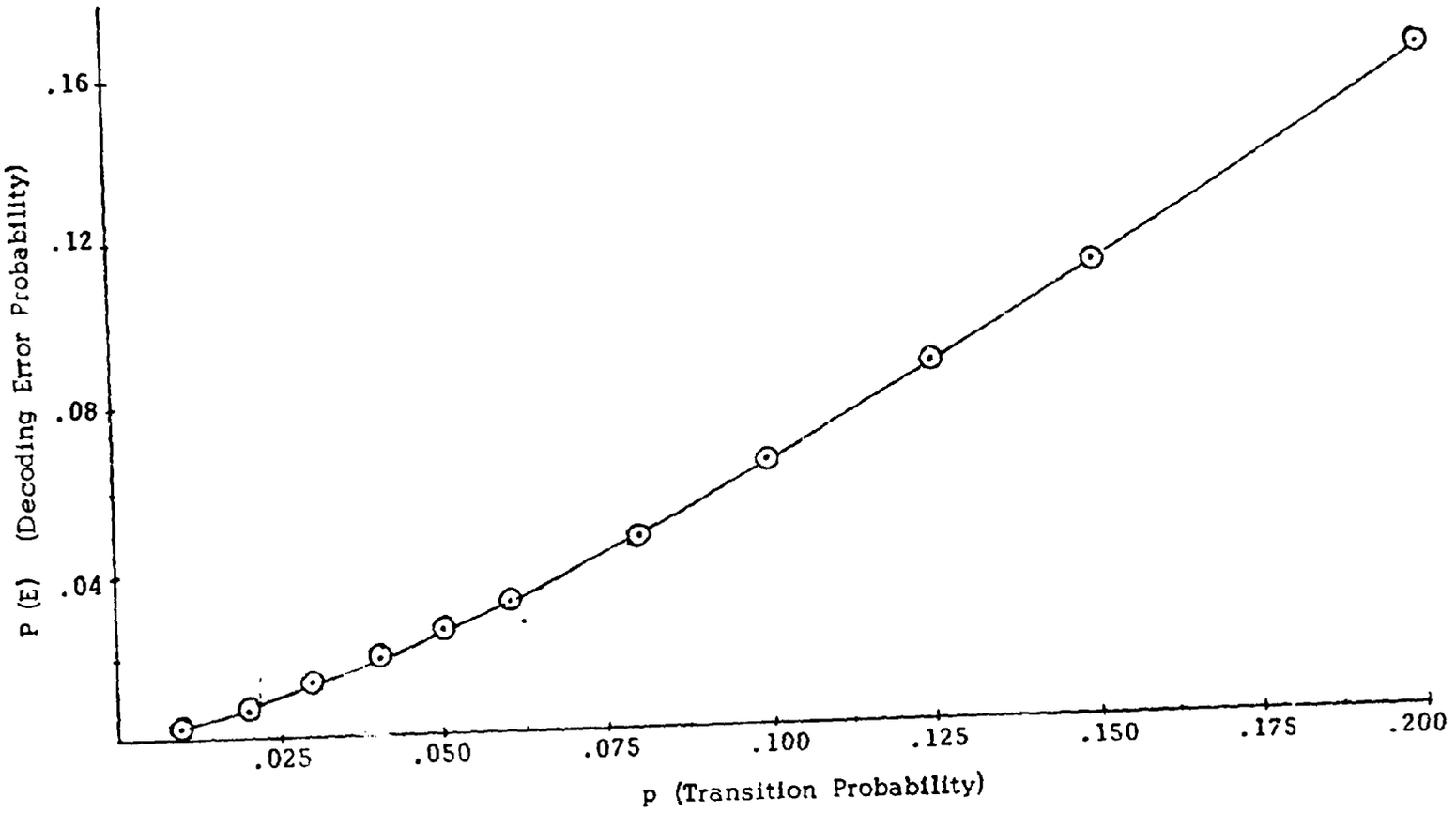

Figure 11. P(E) vs p (P = h = 0.5)

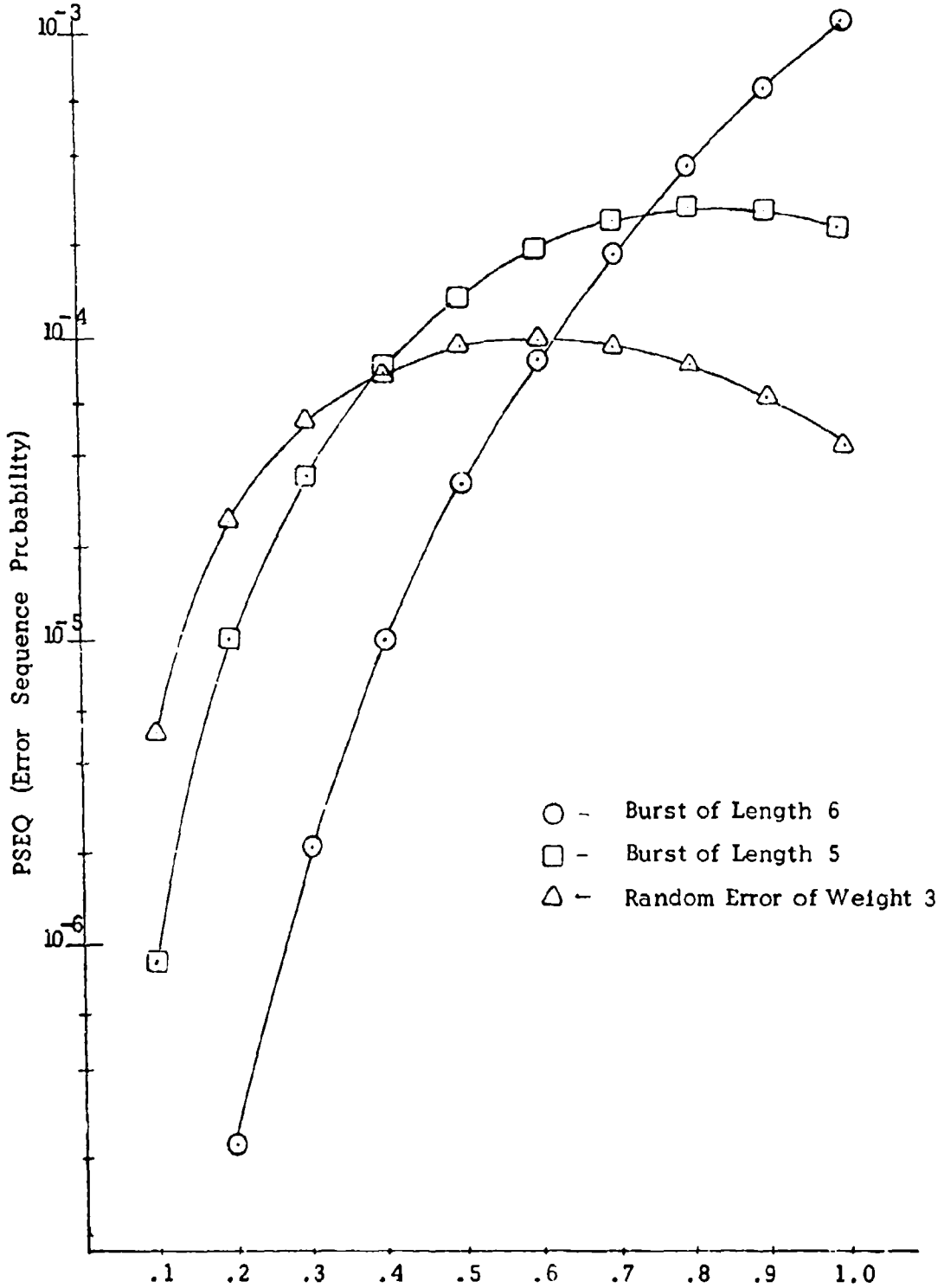

Figure 12. PSEQ vs h (p = .10, P = .50)

LIST OF REFERENCES

1. Shannon, C. E., "A Mathematical Theory of Communication," Bell System Technical Journal, v. 27, p. 379-423, 1948.

2. Peterson, W. W., "Error Correcting Codes," 1961.

3. Gallager, R. G., "Information Theory and Reliable Communication," 1968.

4. Forney, G. D., "Burst-Correcting Codes for the Classic Bursty Channel," IEEE Transactions on Communications Technology, v. 19, October 1971.

5. Gilbert, E. N., "Capacity of a Burst-Noise Channel," Bell System Technical Journal, v. 39, p. 1253-1266, September 1960.

6. Adoul, J. A., Fritchman, B. D., and Kanal, L. N., "A Critical Statistic for Channels with Memory," IEEE Transactions on Information Theory, v. 18, p. 133-141, January 1972.

7. Fritchman, B. D., "A Binary Channel Characterization using Partitioned Markov Chain," IEEE Transactions on Information Theory, v. 13, p. 221-236, April 1967.

8. Gilbert, E. N., "A Comparison of Signaling Alphabets," Bell System Technical Journal, v. 31, p. 504-522, 1952.

9. Plotkin, M., "Binary Codes with Specified Minimum Distance," IRE Transactions, v. 6, p. 445-450, 1960.